IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Craig William Fellenstein	§	Confirmation No.: 2482
	§	
Serial Number: 10/765,777	§	
	§	Group Art Unit: 2166
Filed: 01/27/2004	§	
	§	
For: SYSTEM AND METHOD FOR AUTONOMIC PERFORMANCE ENHANCEMENT OF STORAGE MEDIA	§	Examiner: Navneet K. Ahluwalia
	§	

Commissioner of Patents and Trademarks
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPLICANTS' APPEAL BRIEF

Applicant-inventor ("Applicant") and assignee International Business Machines Corporation respectfully submit the present brief in support of the patentability of the claims of the above-referenced application.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, of Armonk, New York, assignee of the interests in the invention from the named inventor.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

1-8, 10-11, and 13 are pending. Of these, Claims 1, 5, 8, 10, 11, and 13 are independent Claims. Claims 9 and 12 have been canceled. Applicant appeals the Examiner's rejections of Claims 1-8, 10, 11, and 13 under 35 U.S.C. §102(e).

IV. STATUS OF AMENDMENTS

The Claims stand as amended in the Response to an Office Action dated December 6, 2006 ("First Response").

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention under examination is directed to improving the storage algorithms on storage media. *See* Application, Page 1, lines 9-10. Most storage media (*e.g.*, hard disk drives) are write-optimized. *See* Application, Page 1, lines 24-25. They are designed to get the data to the disk as efficiently as possible. *See* Application, Page 1, lines 25-27. In practice, this means that the drives will often break a file into fragments, and use all available heads to write the fragments to different platters simultaneously. *See* Application, Page 1, line 27 to Page 2, line 1. This often results in files being broken into non-contiguous segments on disk, which causes diminished read performance. *See* Application, Page 2, lines 2-13. Periodic defragmentation helps reduce this problem. *See* Application, Page 2, lines 14-16. But typical periodic defragmentation is usually only done when the disk performance is so bad as to be noticeable.

See Application, Page 2, lines 18-20. Periodic defragmentation therefore causes problematic productivity delays. *See* Application, Page 2, lines 20-21.

The present invention, defined in Claims 1-8, 10-11, and 13, solves this problem by providing a novel system and method for autonomic performance enhancement of storage media. Specifically, the present invention introduces a "fragmentation tracker" (FT) and a "defragmentation agent" (DA). *See* Application, Page 5, lines 19-24. The FT monitors and maintains an account of the location of fragmented file clusters. *See* Application, Page 5, line 24 to Page 6, line 2. The DA defragments the files identified by the FT. *See* Application, Page 6, lines 2-4. More specifically, the FT monitors the storage media for scan/write/delete operations that cause file fragmentation, storing the location of fragmented data clusters and ignoring operations that do not cause fragmentation. *See* Application, Page 6, lines 6-18. The DA defragments those files identified by the FT, based on non-intrusive defragmentation operations, defragmenting files only when the system is idle. *See* Application, Page 7, lines 1-17.

This novel approach offers several significant advantages that are unavailable to the prior art methods or systems. First, this approach does not completely lock out disk usage once defragmentation begins. *See* Application, Page 7, lines 9-10. Second, the DA operates whenever the system is idle, instead of waiting until fragmentation causes substantial or critical delays. *See* Application, Page 6, lines 22-24. Third, the DA does not require a user to initiate defragmentation. *See* Application, Page 6, lines 20-21. Other benefits are enumerated throughout the Application.

The Claims embody the invention as follows, shown with illustrative citations to page and line numbers in the Original Application designated in curved braces ("{}"):

1. (Original) An apparatus for file defragmentation of at least one storage medium, comprising:

a computer system at least coupled to the at least one storage medium; {Page 5, Lines 1-5}

a tracker, wherein the tracker is at least configured to maintain a record of at least locations of a plurality of file fragments on at least one storage medium; {Page 5, Line 25 – Page 6, Line 2}and

an agent, {Page 6, Line 2}wherein the agent is at least:

configured to operate while the computer system is at least idle; {Page 7, Lines 1-5}

configured to defragment the plurality of file fragments; {Page 7, Lines 4-5} and

configured to delete the record of at least locations of the plurality of file fragments {Page 7, Lines 7-8}.

5. (Original) An apparatus for file defragmentation of at least one storage medium at least coupled to a computer system, comprising:

a memory, wherein the memory is at least configured to store locations of a plurality of file fragments; {Page 5, Lines 1-5}

an idle monitor, wherein the idle monitor is at least configured to enable defragmentation while the computer system is at least idle; {Page 6, Lines 18-19}

a defragmenter, wherein the defragmenter is at least configured to defragment the plurality of file fragments; {Page 6, Line 2} and

an update monitor, wherein the update monitor is at least configured to delete a record in the memory of at least locations of the plurality of file fragments that at least been defragmented {Page 7, Lines 7-8}.

8. (Original) A method of for file defragmentation of at least one storage medium coupled to a computer system, comprising:

determining if fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage media; {Page 5, Line 25 – Page 6, Line 2}

storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium; {Page 6, Lines 10-14}

determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, defragmenting a file; {Page 7, Lines 4-5}

determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, deleting the location of the fragmented file clusters in the storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, determining if defragmentation is stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, reporting an error. {Page 7, Lines 16-17}

10. (Original) A method of defragmenting at least one storage medium coupled to a computer system, comprising:

determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, defragmenting the file; {Page 7, Lines 4-5}

determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, deleting a location of the fragmented file clusters in a storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, determining if stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, reporting an error. {Page 7, Lines 16-17}

11. (Original) A computer program product for file defragmentation of at least one storage medium at least coupled to a computer system, the computer program product having a medium embodied thereon, the computer program comprising:

computer code for determining if fragmentation occurrs when data is written to, deleted from, or scanned from the at least one storage media; {Page 5, Line 25 – Page 6, Line 2}

computer code for storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium; {Page 6, Lines 10-14}

computer code for determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, computer code for sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, computer code for defragmenting a file; {Page 7, Lines 4-5}

computer code for determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, computer code for deleting the location of the fragmented file clusters in the storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, computer code for determining if defragmentation is stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, computer code for sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, computer code for reporting an error. {Page 7, Lines 16-17}

13. (Original) A computer program product for defragmenting at least one storage medium coupled to a computer system, the computer program product having a medium embodied thereon, the computer program comprising:

computer code for determining if the computer system is idle; {Page 6, Lines 18-19}

if the computer system is not idle, computer code for sleeping for an interval; {Page 7, Lines 1-3}

if the computer system is idle, computer code for defragmenting a file; {Page 7, Lines 4-5}

computer code for determining if defragmentation is complete; {Page 7, Lines 5-6}

if defragmentation is complete, computer code for deleting a location of the fragmented file clusters in a storage medium; {Page 7, Lines 7-8}

if defragmentation is not complete, computer code for determining if stopped by activity; {Page 7, Lines 13-14}

if defragmentation is stopped by activity, computer code for sleeping for an interval; {Page 7, Lines 17-19} and

if defragmentation is not stopped by activity, computer code for reporting an error. {Page 7, Lines 16-17}

VI. GROUNDS OF REJECTION TO BE REVIEWED

Whether Claims 1-8, 10-11, and 13 are patentable over Jochemsen et al. (US 6,757,804 B2).

VII. ARGUMENT

A. Grouping of Claims

Claims 1, 5, 8, 10, 11, and 13 are independent. For purposes of this appeal, Applicant considers each of the independent Claims, and their respective dependent Claims, as separate groups. Thus, the groups of Claims are 1-4, 5-7, 8, 10, 11, and 13.

B. Summary of Pertinent Prosecution

The present application was filed on January 27, 2004, with 13 claims.

The Examiner mailed the First Office Action on September 6, 2006 ("First Action"), rejecting Claims 1-13 under 35 U.S.C. §102(e) as allegedly anticipated by U.S. Patent 6,757,804 by Jochemsen et al. ("Jochemsen").

Applicant responded to the first Office Action on December 6, 2006 ("First Response"), amending Claim 4 to depend from Claim 1 and cancelling Claims 9 and 12. The First Response also argued that Jochemsen failed to teach every element of the Claims. Specifically, Applicant noted that Jochemsen does not teach file defragmentation or operation when the system is idle. *See* First Response, Pages 8-9.

The Examiner mailed the Final Action under appeal on March 7, 2007 ("Final Action"), maintaining the rejections in the First Action.

Applicant requested a telephone interview with the Examiner, which was conducted on May 16, 2007 ("Interview"), with the Examiner's supervisor. During the interview, Applicant enumerated additional limitations not shown in Jochemsen, particularly the steps of Claim 8, including "determining if defragmentation is complete," "if defragmentation is not complete, determining if defragmentation is stopped by activity," "determining if the computer system is idle," and others. The Examiner requested Applicant to provide the discussed argument in a written response after final.

Applicant responded to the Final Action on June 7, 2007 ("Second Response") detailing the shortcomings of the reference as prior art under Section 102, as discussed in the Interview, and reiterating the arguments presented in the First Response.

On June 26, 2007, the Examiner mailed an Advisory Action maintaining the rejections of Claims 1-8, 10, 11, and 13, without addressing the numerous missing elements in the reference. This appeal followed.

C. The Examiner's Rejections Were Procedurally and Factually in Error

1. The Form and Content of the Examiner's Rejections under Section 102 Were Improper and Insufficient

a. Legal Requirements for an Anticipation Rejection

The applicable statute, 35 U.S.C. §102(e), provides, in pertinent part:

> A person shall be entitled to a patent unless –
> (e) the invention was described in - (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for the purposes of this subsection of an application filed in

> the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language

To anticipate a Claim under Section 102, the cited reference must teach each and every element of the Claim. *See* Manual of Patent Examining Procedure (MPEP) §2131. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Further, "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Finally, the claimed elements must be arranged as recited by the claim, but "identity of terminology" is not required. *See In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

In some instances, more than one reference may be employed to support a Section 102 rejection:

> Normally, only one reference should be used in making a rejection under 35 U.S.C. 102. However, a 35 U.S.C. 102 rejection over multiple references has been held to be proper when the extra references are cited to:
>
> (A) Prove the primary reference contains an "enabled disclosure;"
>
> (B) Explain the meaning of a term used in the primary reference; or
>
> (C) Show that a characteristic not disclosed in the reference is inherent.

MPEP §2131.01. Applicants note, however, that in this case, the Examiner has employed only one reference under Section 102(e). Accordingly, the sole Section 102 reference in this case must show each and every element of the Claims.

b. The Examiner's Stated Grounds Were Insufficient

Applicant respectfully submits that the Examiner has completely failed to establish *prima facie* anticipation for each and every element of the pending Claims.

First, regarding Claim 8, the Examiner asserts that Jochemsen teaches, "determining if the computer system is idle if the computer system is not idle, sleeping for an interval." Final Action, Page 7 (*citing* Jochemsen, col. 1, lines 49-58). Nowhere does Jochemsen disclose "determining if the computer system is idle" or performing any actions based on that determination. The Examiner's citation does not even hint at "determining if the computer system is idle":

> When a delete operation is necessary, it is desirable permanently to delete the minimum number of files so that (a) at least the required amount of free space is created; and (b) defragmentation and free-space defragmentation are optimized.
>
> It is an aim of preferred embodiments of the present invention to provide a method, system and corresponding computer program product for reducing fragmentation of a digital storage device.

Jochemsen, col. 1, lines 49-58. As shown, the cited passage does not come anywhere close to teaching any determination of whether the computer system is idle. Applicant respectfully submits that nowhere else does Jochemsen even hint at "determining if the computer system is idle," nor has the Examiner provided any citation whatsoever that does teach this element. Accordingly, as Jochemsen does not show "determining if the computer system is idle" it is therefore not a single prior art reference showing each and every element of Claim 8. As such, Jochemsen cannot support *prima facie* anticipation, and the Examiner's rejection under Section 102 fails for this reason alone.

The Examiner purports to address this argument in the Advisory Action:

> Applicant argues tha [sic] Jochemsen does not teach determining id [sic] the computing system is not idle. The argument is respectfully traversed as it requires the agent to operate at least when the system is idle which is clearly disclosed in Jochemsen. Furthermore, Jochemsen discloses in column 2 lines 1 – 10 and 58 – 67 and column 3 lines 23 – 45 the operation of the agent in the computer system irrespective of it being idle or non-idle.

Advisory Action, Page 3. The Examiner's assessment completely misses the point. Notwithstanding the Examiner's assertions that Jochemsen operates "irrespective of it being idle or non-idle," Jochemsen nowhere makes a *determination* whether the system is idle, as recited in the claim. Final Action, Page 2. In fact, the Examiner's position illustrates why Jochemsen does not teach a determination whether the system is idle: according to the Examiner, Jochemsen operates irrespective of the system's idle state. But even if the Jochemsen system does *operate* when idle, which Applicant does not admit, it still must teach *determining* whether the system is idle (as well as the other missing elements) in order to sustain a proper anticipation rejection. Nowhere does Jochemsen even hint at this teaching.

Jochemsen likewise fails to teach a number of other elements of Claim 8, as well. For example, nowhere does Jochemsen teach "determining if defragmentation is complete [sic] if defragmentation is complete, deleting the location of the fragmented file clusters in the storage medium" as the Examiner alleges. Final Action, Page 7 (*citing* Jochemsen, col. 1, lines 49-53). As shown above, the cited reference does not teach anything close to "determining if defragmentation is complete." *See supra*; Jochemsen, col. 1, lines 49-53. Applicant respectfully submits that nowhere else does Jochemsen teach "determining if defragmentation is complete."

Neither does Jochemsen teach, "if defragmentation is not complete, determining if defragmentation is stopped by activity" as the Examiner alleges. Final Action, Page 7 (*citing* Jochemsen, col. 3, lines 22-34). The cited passage states:

> When a deletion operation is initiated (step 200) the file manager 30 is interrogated (step 202) to determine whether multiple files are available for deletion. A file is available for deletion if it has been so annotated. A file is only available for deletion if in doing so it provides sufficient free-space for the required write operation. This may mean that several of a plurality of files need to be deleted. If only a single file is available for deletion, it is deleted (step 204) and the file manager 30 updated accordingly (step 210). If multiple files are

> available a file is selected to be deleted to reduce fragmentation (step 206). Options for selecting the file to be deleted in this embodiment of the present invention are set out below. Next the selected file is permanently deleted (step 208) and the file manager 30 is updated accordingly (step 210).

Jochemsen, col. 3, lines 22-36. As shown, nowhere in the cited passage does Jochemsen even hint at "stopping defragmentation" much less "determining whether defragmentation is stopped by activity," as recited in Claim 8. Applicant respectfully submits that nowhere else does Jochemsen even hint at "determining whether defragmentation is stopped by activity." Accordingly, Jochemsen fails to show this element, as well.

Applicant respectfully submits that, similarly, Jochemsen also fails to show "sleeping for an interval" or "reporting an error" as a function of "determining if defragmentation is stopped by activity." These two elements are also missing from Jochemsen. For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102 rejection of Claim 8, as Jochemsen completely fails to teach several elements of Claim 8. Accordingly, Applicant respectfully submits that the Section 102 rejection of Claim 8 is not supported by the reference, and is therefore completely improper.

Next, Applicant respectfully submits that Jochemsen also fails to teach each and every element of the remaining pending Claims, in a similarly clear manner as above. Specifcally, the Examiner's cited passages wholly fail to even mention, much less support the elements they are offered to teach. For example, regarding Claim 10, Applicant respectfully submits that Jochemsen fails to teach at least the following elements recited in Claim 10, "determining if the computer system is idle", "determining if defragmentation is complete", "determining if [defragmentation] is stopped by activity" and "sleeping for an interval" or "reporting an error"

based on whether defragmentation is stopped by activity. As described above, these elements are wholly missing from Jochemsen.

Regarding Claim 10, the Examiner cites a different passage to support "determining if the computer system is idle," than the passage offered against Claim 8, but that passage also teaches nothing remotely close to "determining if the computer system is idle." *See* Final Action, Page 8 (*citing* Jochemsen, col. 2, lines 1-10). Specifically, the cited passage states:

> According to the present invention in a second aspect, there is provided a system arranged for reducing fragmentation of a digital storage device, characterized by the system comprising means for determining that a plurality of files is available for deletion; means for selection one of the files; and means for deleting the selected file and not deleting another of the files.
>
> It has been realized that with digital storage devices annotating a plurality of files for deletion, there is an efficient opportunity to reduce fragmentation by selective file deletion.

Jochemsen, col. 2, lines 1-11. Clearly, the cited passage comes nowhere near teaching "determining whether the computer system is idle." For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102 rejection of Claim 10, as Jochemsen also completely fails to teach several elements of Claim 10. Accordingly, Applicant respectfully submits that the Section 102 rejection of Claim 10 is not supported by the reference, and is therefore completely improper.

Next, regarding Claim 11, Applicant respectfully submits that Jochemsen also fails to teach at least the following elements recited in Claim 11, "determining if the computer system is idle", "determining if defragmentation is complete," "determining if [defragmentation] is stopped by activity" and "sleeping for an interval" or "reporting an error" based on whether defragmentation is stopped by activity. As described above, these elements are wholly missing from Jochemsen. For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102

rejection of Claim 11, as Jochemsen completely fails to teach several elements of Claim 11. Accordingly, Applicant respectfully submits that the Section 102 rejection of Claim 11 is not supported by the reference, and is therefore completely improper.

Next, regarding Claim 13, Applicant respectfully submits that Jochemsen fails to teach at least the following elements recited in Claim 13, "determining if the computer system is idle", "determining if defragmentation is complete," "determining if [defragmentation] is stopped by activity" and "sleeping for an interval" or "reporting an error" based on whether defragmentation is stopped by activity. As described above, these elements are wholly missing from Jochemsen. For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102 rejection of Claim 13, as Jochemsen completely fails to teach several elements of Claim 13. Accordingly, Applicant respectfully submits that the Section 102 rejection of Claim 13 is not supported by the reference, and is therefore completely improper.

Next, regarding Claims 3 and 7, the missing elements are even more blatant. Applicant respectfully submits that Jochemsen also fails to teach modifying the attributes of defragmentation "wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access," as asserted by the Examiner. Final Action, Pages 4 and 6 (*citing* Jochemsen, col. 4, lines 22-226 [sic], and 42-47). Applicant respectfully submits that nowhere does Jochemsen, either in the cited passages or elsewhere, recite the above limitation.

For example, Jochemsen recites:

> The determination of which file to delete whichever selection option is chosen is carried out by interrogating the file manager 30. For every file in a set of

> deletable files the effect of deletion on file fragmentation and free-space fragmentation is calculated.
>
> . . .
>
> The effect on free-space fragmentation is calculated by calculating the changes the deletion (for each file) would have on the free-space fragments (step 402). New free-space fragments might appear (undesirable), but several free-space fragments can be connected by deletion creating new free-space fragments (desirable). In general, the number of free-space fragments will increase (i.e. a positive change).

Jochemsen, col. 4, lines 22-26, 42-47. Clearly, this passage does not teach "wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access," as recited in Claims 3 and 7. Applicant respectfully submits that nowhere else does Jochemsen teach this element.

For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102 rejection of Claims 3 and 7, as Jochemsen completely fails to teach several elements of Claims 3 and 7. Accordingly, Applicant respectfully submits that the Section 102 rejections of Claims 3 and 7 are not supported by the reference, and are therefore completely improper.

Next, regarding Claims 2 and 6, Applicant respectfully submits that Jochemsen similarly fails to teach "wherein the agent further comprises at least having the ability to modify attributes of defragmentation," as asserted by the Examiner. Final Action, Pages 4 and 6 (*citing* Jochemsen, col. 3, lines 37-46). Applicant respectfully submits that nowhere does Jochemsen, either in the cited passages or elsewhere, recite the above limitation.

Instead, the cited passage teaches selecting one of a number of files for deletion, stating, "there can be several criteria to determine the text file to delete." Jochemsen, col. 3, lines 40-41. Applicant respectfully submits that this fails to teach "wherein the agent further comprises at

least having the ability to modify attributes of defragmentation," as recited in Claims 2 and 7. Applicant respectfully submits that nowhere else does Jochemsen teach this element.

For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102 rejection of Claims 2 and 6, as Jochemsen completely fails to teach several elements of Claims 2 and 6. Accordingly, Applicant respectfully submits that the Section 102 rejections of Claim 2 and 6 are not supported by the reference, and are therefore completely improper.

Finally, regarding Claim 1, the Examiner cites Jochemsen as teaching, "an agent, wherein the agent is at least: configured to operate while the computer system is at least idle (column 2 lines 58-67, Jochemsen); configured to defragment the plurality of file fragments (column 2, lines 1-10, Jochemsen); and configured to delete the record of at least locations of the plurality of file fragments (column 2 lines 58-67 and column 3 lines 23-45)." Final Action, Page 3.

The Examiner cites column 2, lines 1-10, of Jochemsen as teaching defragmentation. However, the actual text, reproduced above, instead teaches "reducing fragmentation" not active defragmentation. Applicant respectfully submits that Jochemsen teaches deletion and writing in a manner to reduce fragmentation of a file *before* the file is written to a disk. Conventional defragmentation takes place *after* a series of files have been written to a disk. In Jochemsen, no individual clusters are moved or rearranged to form more contiguous units. Instead, Jochemsen is essentially a specialized delete function in which files to be deleted are analyzed along with the fragmented free space and the size of a write file in order to delete only those files necessary to place the write file on the disk in a manner that reduces the overall fragmentation of the disk. *See* Jochemsen, col. 2, lines 51-58. A user implementing the teachings of Jochemsen may still have to actively defragment the disk in order to optimize performance.

Moreover, Claim 1 includes an agent "at least: configured to operate while the computer system is at least idle." The Examiner cites column 2, lines 58-67 for support of anticipation of this element. As described above, there is no determination whatsoever in Jochemsen of whether the computer system is or is not idle. Accordingly, Applicant respectfully submits that nowhere does Jochemsen teach this element, either.

As described above, Jochemsen teaches marking a file for deletion, whether the file is marked immediately or subsequently. The Jochemsen deletion takes place during periods of computer activity in which the computer system is *actively* instructed to write a file or clear a space. Nowhere does Jochemsen teach determination that the computer system is idle, or any actions associated therewith, as described above.

For at least the above reasons, Applicant has shown that Jochemsen cannot support a Section 102 rejection of Claim 1, as Jochemsen completely fails to teach several elements of Claim 1. Accordingly, Applicant respectfully submits that the Section 102 rejection of Claim 1 is not supported by the reference, and is therefore completely improper.

Similarly, Claim 5 recites, "an idle monitor, wherein the idle monitor is at least configured to enable defragmentation while the computer system is at least idle." Applicant respectfully submits that Jochemsen therefore fails to teach each and every element of Claim 5, for the same reasons as described above regarding Claim 1. As such, neither can Jochemsen cannot support a Section 102 rejection of Claim 5, as Jochemsen completely fails to teach several elements of Claim 5. Accordingly, Applicant respectfully submits that the Section 102 rejection of Claim 5 is not supported by the reference, and is therefore completely improper.

Additionally, dependent Claims 2-4, 6, and 7, depend upon and further limit independent Claims 1 and 5. Hence, for at least the aforementioned reasons, these dependent Claims would be deemed to be in condition for allowance. Therefore, Applicant also respectfully submits that the Section 102 rejections of Claims 2-4, 6, and 7, are not supported by the reference, and are therefore completely improper.

Applicant has now positively demonstrated the pervasive failure of the Examiner's reference to teach each and every element recited in the Claims. The failure of the Examiner's only reference to teach each and every element is fatal to the Section 102 rejections, which must therefore be overturned. As the Examiner's only rejections of the Claims are under Section 102, Applicant respectfully submits that the Claims now stand in full condition for allowance. Applicant therefore respectfully requests full allowance of Claims 1-8, 10-11, and 13.

VIII. CLAIMS APPENDIX

See Attached.

IX. EVIDENCE APPENDIX

NONE.

X. RELATED PROCCEDINGS APPENDIX

NONE.

XI. CONCLUSION

For the foregoing reasons, it is respectfully submitted that the Final Rejections of Claims 1-8, 10-11, and 13 under 35 U.S.C. §102(e) are improper and should be reversed. Applicants respectfully request that the rejections of Claims 1-8, 10-11, and 13 be withdrawn and that Claims 1-8, 10-11, and 13 be allowed.

Respectfully submitted,

Dated: October 9, 2007

/Gregory W. Carr/
Gregory W. Carr
Reg. No. 31,093

CARR LLP
670 Founders Square
900 Jackson Street
Dallas, Texas 75202
Telephone: (214) 760-3032
Fax: (214) 760-3003

VIII – APPENDIX – CLAIMS ON APPEAL

1. (Original) An apparatus for file defragmentation of at least one storage medium, comprising:

a computer system at least coupled to the at least one storage medium;

a tracker, wherein the tracker is at least configured to maintain a record of at least locations of a plurality of file fragments on at least one storage medium; and

an agent, wherein the agent is at least:

configured to operate while the computer system is at least idle;

configured to defragment the plurality of file fragments; and

configured to delete the record of at least locations of the plurality of file fragments.

2. (Original) The apparatus of Claim 1, wherein the agent further comprises at least having the ability to modify attributes of defragmentation.

3. (Original) The apparatus of Claim 2, wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access.

4. (Previously Presented) The apparatus of Claim 1 further comprising:

a memory, wherein the memory is at least configured to store locations of a plurality of file fragments;

a system monitor, wherein the system monitor at least determines if file fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage medium; and

an accounting means, wherein the accounting means is at least configured to store locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred.

5. (Original) An apparatus for file defragmentation of at least one storage medium at least coupled to a computer system, comprising:

a memory, wherein the memory is at least configured to store locations of a plurality of file fragments;

an idle monitor, wherein the idle monitor is at least configured to enable defragmentation while the computer system is at least idle;

a defragmenter, wherein the defragmenter is at least configured to defragment the plurality of file fragments; and

an update monitor, wherein the update monitor is at least configured to delete a record in the memory of at least locations of the plurality of file fragments that at least been defragmented.

6. (Original) The apparatus of Claim 5, wherein the agent further comprises at least having the ability to modify attributes of defragmentation.

7. (Original) The apparatus of Claim 6, wherein the attributes are selected from the group consisting of file type, frequency of access, typical access duration, interval between accesses, file/application association, file size, read attributes, update attributes, and time of day of typical access.

8. (Original) A method of for file defragmentation of at least one storage medium coupled to a computer system, comprising:

determining if fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage media;

storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium;

determining if the computer system is idle;

if the computer system is not idle, sleeping for an interval;

if the computer system is idle, defragmenting a file;

determining if defragmentation is complete;

if defragmentation is complete, deleting the location of the fragmented file clusters in the storage medium;

if defragmentation is not complete, determining if defragmentation is stopped by activity;

if defragmentation is stopped by activity, sleeping for an interval; and

if defragmentation is not stopped by activity, reporting an error.

9. (Canceled).

10. (Original) A method of defragmenting at least one storage medium coupled to a computer system, comprising:

determining if the computer system is idle;

if the computer system is not idle, sleeping for an interval;

if the computer system is idle, defragmenting the file;

determining if defragmentation is complete;

if defragmentation is complete, deleting a location of the fragmented file clusters in a storage medium;

if defragmentation is not complete, determining if stopped by activity;

if defragmentation is stopped by activity, sleeping for an interval; and

if defragmentation is not stopped by activity, reporting an error.

11. (Original) A computer program product for file defragmentation of at least one storage medium at least coupled to a computer system, the computer program product having a medium embodied thereon, the computer program comprising:

computer code for determining if fragmentation occurrs when data is written to, deleted from, or scanned from the at least one storage media;

computer code for storing locations of a plurality of file fragments when the system monitor has at least determined that file fragmentation has occurred in a storage medium;

computer code for determining if the computer system is idle;

if the computer system is not idle, computer code for sleeping for an interval;

if the computer system is idle, computer code for defragmenting a file;

computer code for determining if defragmentation is complete;

if defragmentation is complete, computer code for deleting the location of the fragmented file clusters in the storage medium;

if defragmentation is not complete, computer code for determining if defragmentation is stopped by activity;

if defragmentation is stopped by activity, computer code for sleeping for an interval; and

if defragmentation is not stopped by activity, computer code for reporting an error.

12. (Canceled).

13. (Original) A computer program product for defragmenting at least one storage medium coupled to a computer system, the computer program product having a medium embodied thereon, the computer program comprising:

computer code for determining if the computer system is idle;

if the computer system is not idle, computer code for sleeping for an interval;

if the computer system is idle, computer code for defragmenting a file;

computer code for determining if defragmentation is complete;

if defragmentation is complete, computer code for deleting a location of the fragmented file clusters in a storage medium;

if defragmentation is not complete, computer code for determining if stopped by activity;

if defragmentation is stopped by activity, computer code for sleeping for an interval; and

if defragmentation is not stopped by activity, computer code for reporting an error.